opentext™

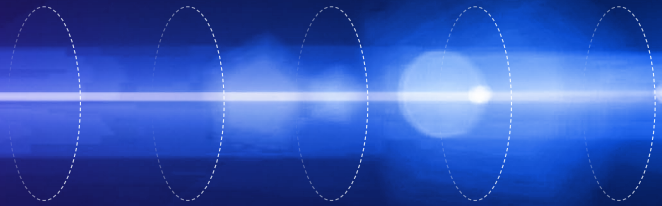
ADM Market Insight:
How to Integrate Open Source and Commercial Tools into Your DevOps Pipeline

Key Takeaway

Many companies use a blend of commercial and open source products—commercial tools to handle core business activities, and open source tools to manage supporting business operations in a cost-effective manner. Find out how you can successfully combine both within your DevOps pipeline to maximize return on investment and

Introduction

There is a principle in enterprise architecture that applies to most organizational decisions—reuse before buy before build. It means you should look at reusing what you have before you consider buying or sourcing something new. And only consider building it yourself when you can't find what you want.



The same principle holds true when we look at the role of development teams and DevOps tools. For most companies, developing a suite of DevOps tools to build products and solutions is not a primary driver. Organizations have other priorities to consider before trying to develop DevOps solutions, particularly since only your DevOps teams may ever see or use them.

Many organizations use a blend of both commercial and open-source products to maximize their return on investment and focus on adding customer value. They might use commercial tools to handle core and critical activities within an organization and rely on open source products (where appropriate) to deliver supporting capabilities. Here we will dive into where commercial and open source tools fit in your enterprise, why you might choose one over the other, and how organizations can combine both to provide maximum flexibility and value for the money.

The Case for Commercial Tools

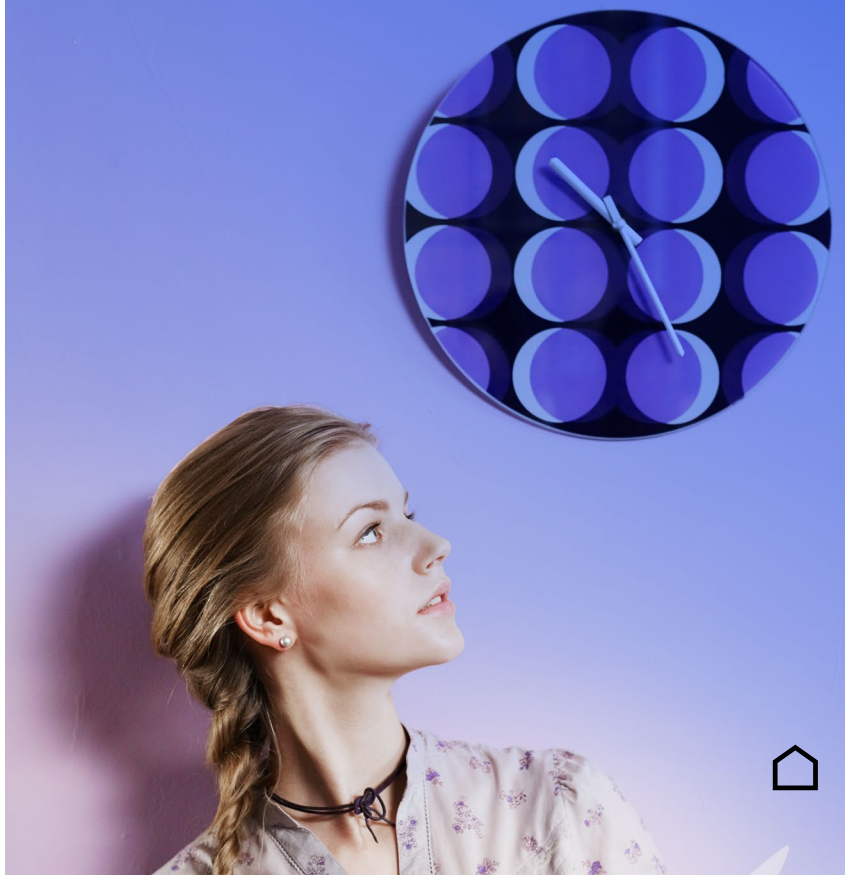
There are several reasons for important business functions to be supported by commercial solutions. In particular, purchasing commercial products enables organizations to have very clear, commercial agreements with the providers of products. This includes explicit, specific licensing and usage requirements for a product. Each purchased product will also have support agreements, and in most cases, a well-defined point of contact.

Commercial contracts and agreements allow organizations to rely on the products and support for day-to-day, critical operations. These products generally have defined upgrade paths and well-communicated development roadmaps that are driven by feedback from commercial clients and allow organizations to plan process changes more consistently.

There are also categories of tools and solutions that are very complex or high risk. For these categories, it makes sense to invest in a commercial solution rather than using open source tools (unless you are delivering this functionality as a product). For example, security and testing tools are the type of software to consider purchasing from a commercial vendor, since they offer significant value to your organization but require substantial time and resources to develop.

Finally, regulatory compliance is a key concern, too. For enterprises that operate in an industry that has specific compliance requirements—such as government contracting or healthcare—purchasing an off-the-shelf solution is often preferable.

Using open source tools may make it necessary to go through lengthy, costly, and often unreliable processes to self-certify the open source code for compliance every time there is a new release.

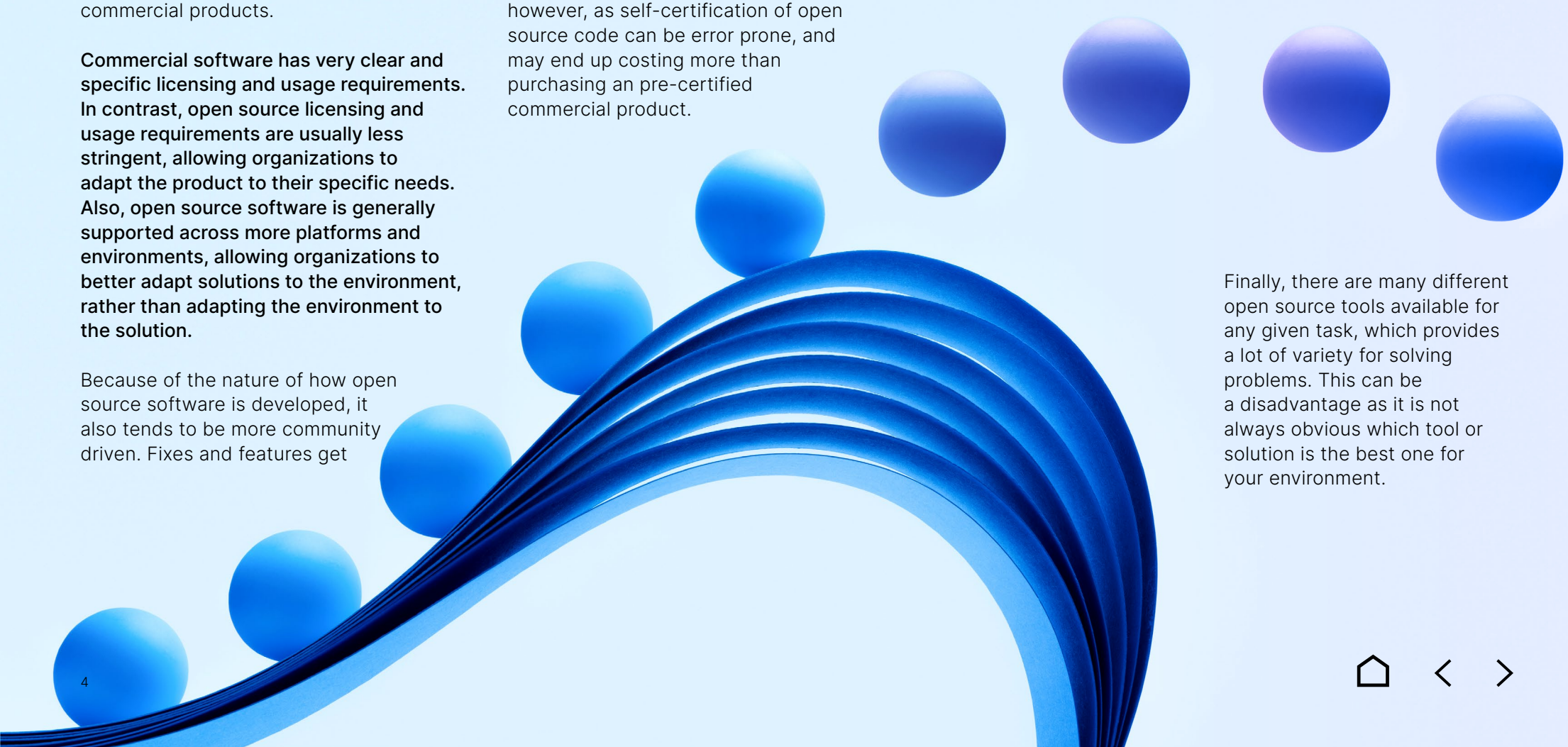


Where Does Open Source Make Sense?

Purchasing a wide variety of commercial software products to support all business functions can be an expensive exercise with a limited return on investment. Leveraging a cost-effective open source solution, especially for supporting or non-critical business functions, can have significant advantages over commercial products.

Commercial software has very clear and specific licensing and usage requirements. In contrast, open source licensing and usage requirements are usually less stringent, allowing organizations to adapt the product to their specific needs. Also, open source software is generally supported across more platforms and environments, allowing organizations to better adapt solutions to the environment, rather than adapting the environment to the solution.

Because of the nature of how open source software is developed, it also tends to be more community driven. Fixes and features get added at a more rapid pace by many contributors across the world, rather than limited to employees supporting a commercial product. Also, because the source and software are open, organizations can audit the code for security and compliance requirements themselves. Beware of auditing costs, however, as self-certification of open source code can be error prone, and may end up costing more than purchasing an pre-certified commercial product.



Finally, there are many different open source tools available for any given task, which provides a lot of variety for solving problems. This can be a disadvantage as it is not always obvious which tool or solution is the best one for your environment.

The Case for Using Both

Organizations can leverage open source software to save money by adopting its capabilities for development-related functions while investing in targeted commercial solutions for the core business. By investing in commercial software, organizations can rely on the tools required to maximize the value to their core business, such as security tools, to increase client confidence in storing data with them. Mixing in open source solutions can help with the scale and rapid change required to support the organization, without the need for large investments that are hard to justify outside of core business functions.

Comparing Open Source and Commercial Tools Services

SERVICES	OPEN SOURCE	COMMERCIAL
Dedicated support	Community only	Full support
Online help/documentation	Forums and community only	Centralized, extensive content
License management	User-based, non-centralized	Centralized, admin-controlled
Security and compliance	Not supported	Supported
Privacy standards	Not supported	Supported
Vendor-driven training	Not supported	Supported
Guaranteed compatibility	Not supported	Supported

This concept also applies to the tools and solutions used by individual teams and employees, as well as organization-wide solutions. Each team has activities it performs that are core to the business, as well as a number of supporting activities, where a blend of commercial and open source tools are commonplace.

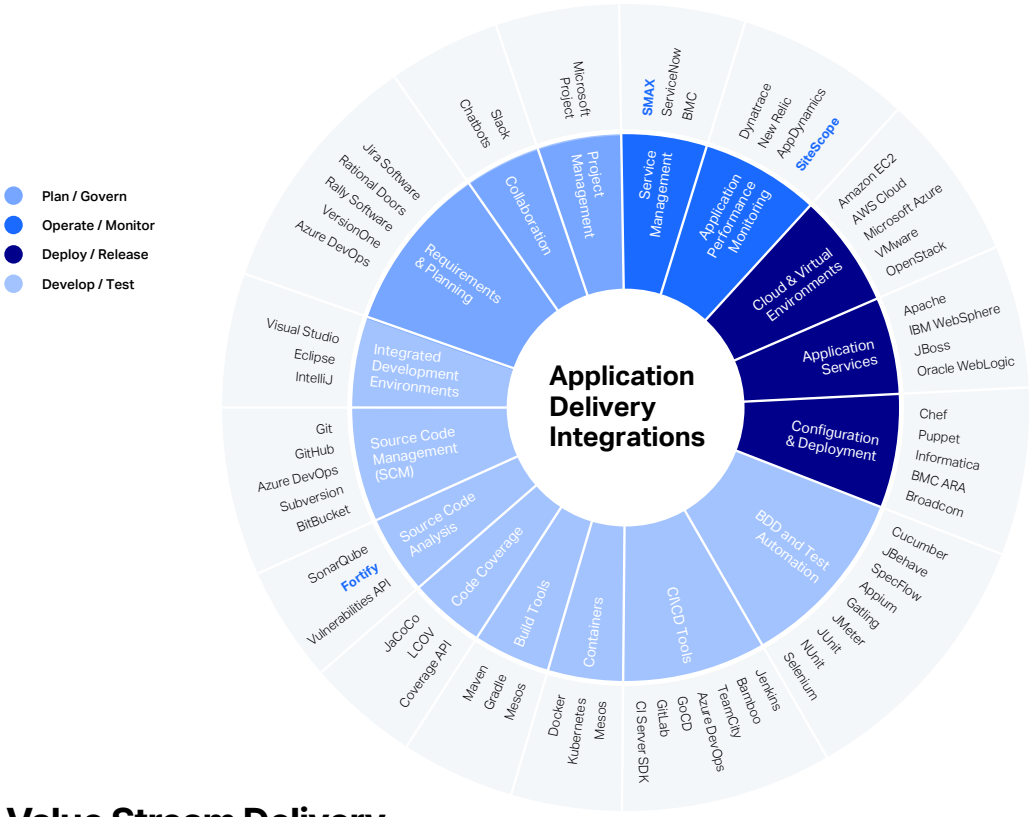
For instance, an organization might invest in OpenText UFT One to provide automated functional and regression testing of GUIs and APIs for testers and test teams working on core business applications. Since robust, reliable tools are critical to the business value testers provide, commercial testing tools are a good investment.

On the other hand, developers working on these same applications may leverage Selenium to test their code at a high level. BDD/TDD test frameworks, like Cucumber, Jasmine, JUnit and NUnit, are also commonly used to run tests before sending it to dedicated QA teams for more rigorous testing on commercial tools.

How to Use Both

Integrating both open source and commercial products can make sense when automating processes, but integration of products should extend past this. Getting a complete picture of the value delivered across these tools is also required. If the tools don't talk to each other this can be quite challenging. This is where integration and aggregation of the data ingested and generated by software is very useful.

OpenText embraces the open source ecosystem and enables customers to leverage commercial tools in partnership with any technology (free or third-party). OpenText provides much needed enterprise capabilities that augment open source to support greater scalability, integration and security, and reduce some of the risk that open source can introduce into the enterprise.



Value Stream Delivery

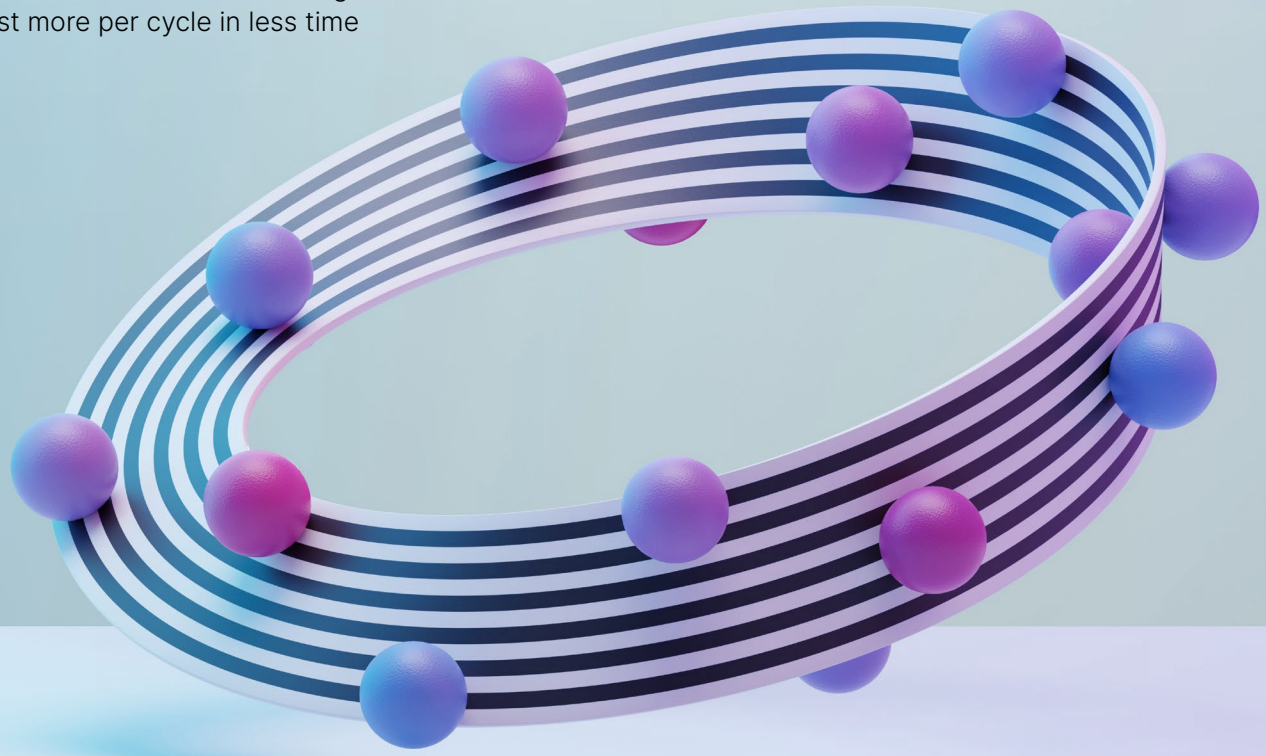
The complexity of building and maintaining connections between disparate toolchains often inhibits a continuous delivery ecosystem and makes it more difficult to trace work items across teams and toolchains. OpenText ALM Octane acts as a central hub that connects people and technology to align work from ideation through release—regardless of the number of teams, projects and locations. With extensive support for open source, teams can:

› Observe trends, identify bottlenecks, find correlations, and detect anomalies in the development cycle
› Incorporate pipeline data into the overall process, helping analyze quality, progress, change impact, code coverage and more
› Manage both manual and automated testing that scales to thousands of tests

Functional Testing

The customer expectation that everything will work everywhere, every time, is driving the rapid pace of change, requiring faster and earlier application testing. **OpenText UFT solutions** emphasize a “shift-everywhere” within a continuous delivery pipeline across testing ecosystem and lab. OpenText enhances open source testing tools with:

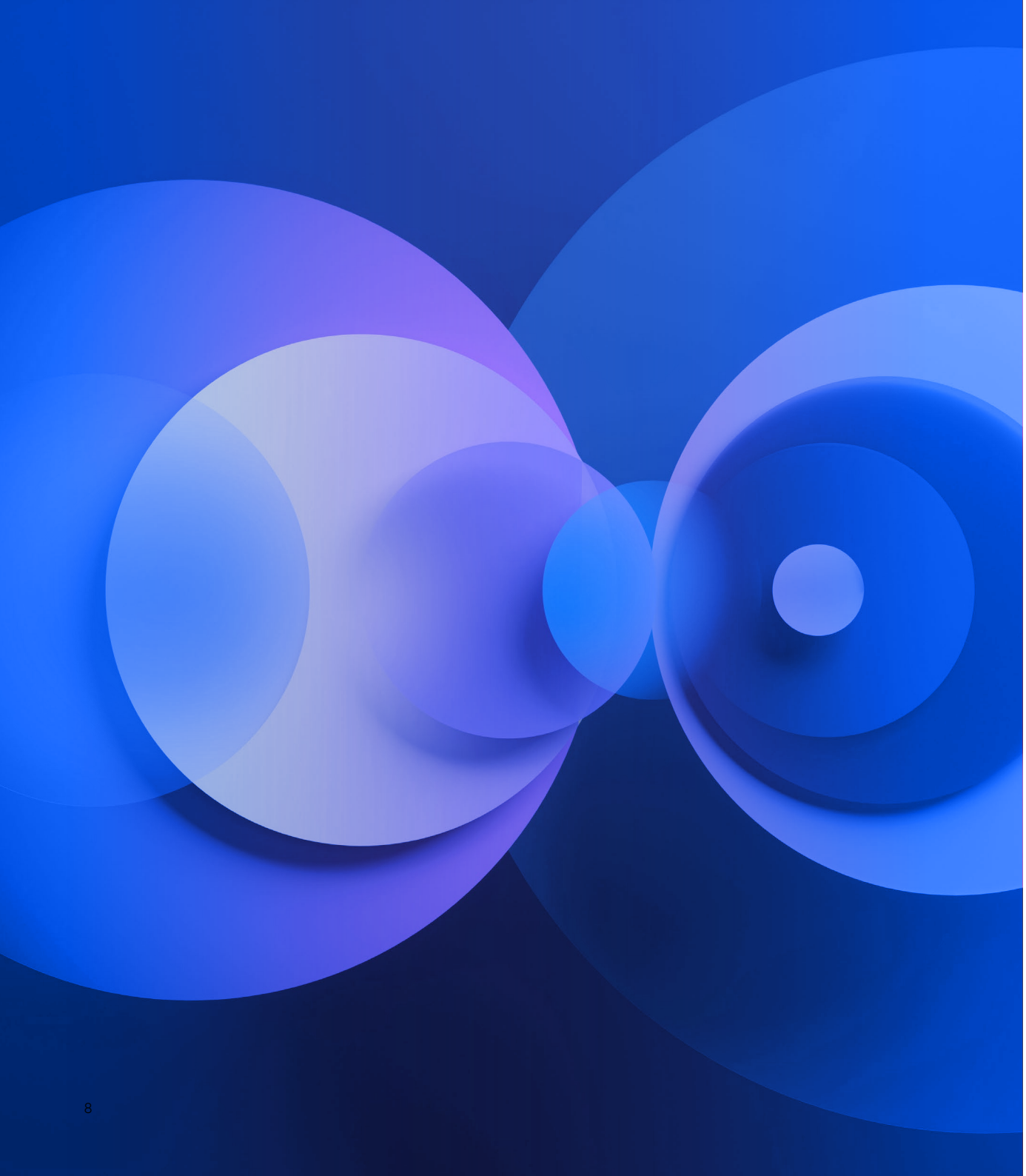
- AI-infused model to increase test coverage and resiliency, and reduce test creation time and maintenance efforts
- Accelerated feedback loops about the quality of application tests at each step
- Extensive support for 200+ enterprise apps, technologies, and environments from a centralized testing solution
- Ability to test more per cycle in less time



Performance Engineering

The domain of performance testing has undergone a fundamental shift from simple load testing to a performance engineering driven approach. This places more emphasis on realistic simulations at greater scale and deep analytics for improving application design during development and faster remediation later on. **OpenText LoadRunner solutions** not only integrate and support open source solutions, but add much needed enterprise capabilities:

- Scalability to 5,000,000 or more geographically distributed web and mobile users
- Real-time monitors with application-layer and code-level data for root cause and analytics
- A globally accessible platform for sharing best practices and resources



Investing Wisely

Many organizations choose to combine open source tools for support functions and commercial tools for critical capabilities. It helps ensure teams and organizations are delivering business value in a cost-effective and reliable way. Making decisions around which type of solutions and tools to invest in is important.

By focusing financial investment in solutions that support core capabilities for teams, and by leveraging open source solutions for support functionality, you can maximize the impact of these investments. The complexity of the functionality involved also needs to be factored in—for instance with testing and security. Decisions around building tools or leveraging open source can have a large impact on your organization.

For solutions that help you build, test, and secure the apps your enterprise needs, check out OpenText Application Delivery Management portfolio for additional information.

Learn more

opentext™

243-000025-001